BUILDING A STRATEGY FOR MORE ACCESSIBLE DESIGN SYSTEMS

Sketch Meetup · Munich · March 25, 2020

WHEELCHAIR RAMP
AVAILABLE
PLEASE ASK AT COUNTER

Hej! 👋
I’m Damien.

Hej! 👋

I’m **Damien.**

I am a **queer Web worker**,
designing **design systems**
and trying to make **accessibility** more accessible.

I work for **Castor** in Amsterdam
as a **Design systems & Accessibility Lead**.

Oh, and my pronouns are **they/them**.

Let’s talk about
crushing dreams.

Let’s talk about frustrations.

In other words, let's talk about design systems & accessibility.

1.

Design systems & accessibility: a reality check.

Photo by Balázs Kétyi on Unsplash

Accessibility is complex.

Design systems will not make accessibility easier or less complex.

A good code can't make a bad design accessible.
A good design can't make a bad content accessible.
A good content can't make a bad idea accessible.

Even with accessible components, you can easily mess-up the end result.

And don't consider fixing accessibility later.
It is always way more expensive. Way more.

Let’s do a test!

- ✍️ Is there any user generated content?
- ♿ Are your teammates trained on accessibility?
- 🎨 How accessible is the brand colour palette?
- 🕹️ Do you have a device lab with assistive techs?
- 🎪 Do you have stakeholders?

You’re doomed.

But relax, we have ~~cookies~~ solutions. 🧘‍♀️

2.

Accessibility is a team effort

I am about to show you the footage of a designer trying to share UX feedback with engineers.

No engineers were harmed in the making of this recording.

I HATE EVERYONE

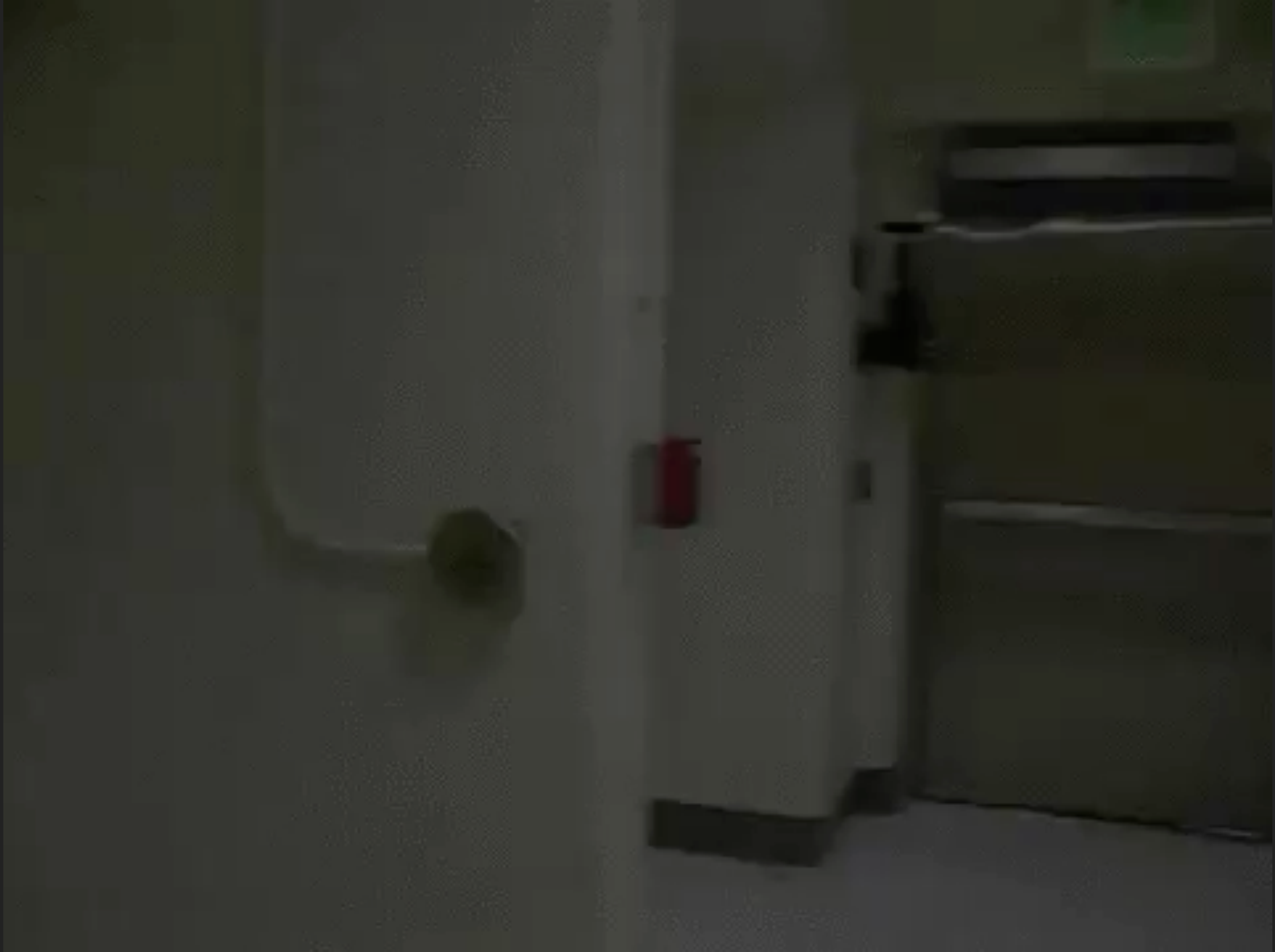
As often, the key to more accessible products is your ability to communicate.

If accessibility is not discussed with everybody, you will start **working in isolation**.

Accessibility is about the global experience.

Accessibility is about details.

Accessibility is about everything.



First, some basic accessibility requirements should be part of your definition of done.

Being part of the definition of done will help your team to estimate the effort including it. (not as an additional package)

You should also align on what does it mean in your context to be accessible.

Accessibility is not just a technical topic.

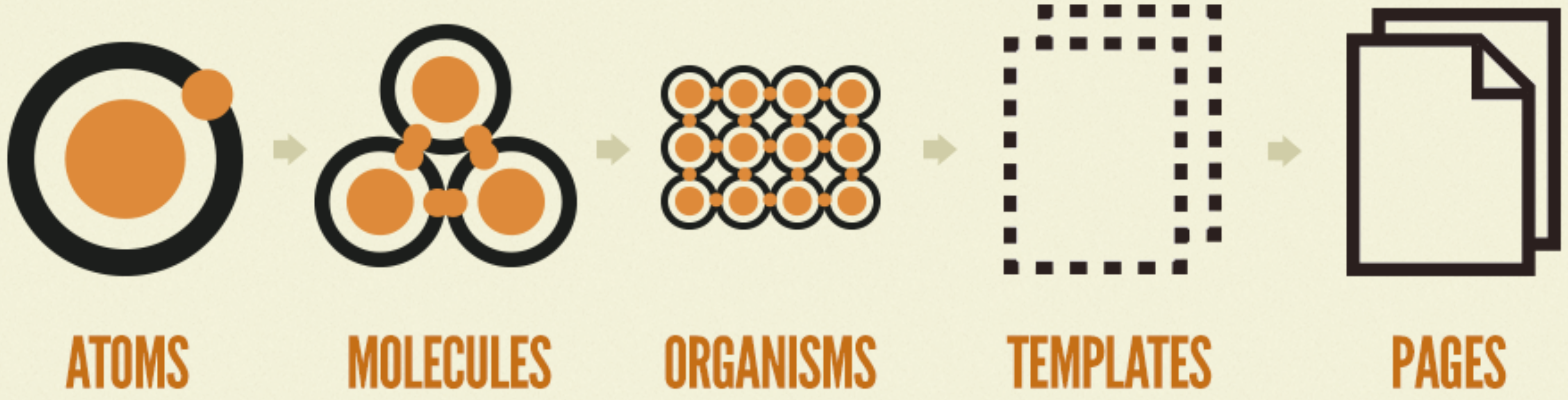
Using accessible components is not enough to offer accessible experiences.

Accessibility is **not** something we can automate.

3.

So what can we do?

Working with the atomic design principles can allow you to be more efficient



Atomic design by Brad Frost

Atoms are perfect to work on semantic.

Molecules are a great place to work what's invisible.

Organisms are the place to be for user focus management and event handling.

Templates can be the higher-level focusing on content order and user navigation.

And for that: we need one important thing.

Please create an environment
where HTML & CSS are valued skills.

HTML & CSS are complex languages,
so create opportunities for people to learn more.

HTML & CSS are also often a good opportunity for designers and engineers to sit together.

Develop a team of accessibility champions with members in all teams and departments.

These evangelists can help finding solutions or mentor colleagues as early as possible.

#shareTheLove

These accessibility champions might not be experts,
but people caring about this topic,
ready to raise the subject when needed.

Develop an environment where people are collaborating around QA tasks.

⌨️ Did you test the keyboard navigation?

🕹️ Did you test it at least one assistive technology?

🤖 Did you run Accessibility Insights for Web?

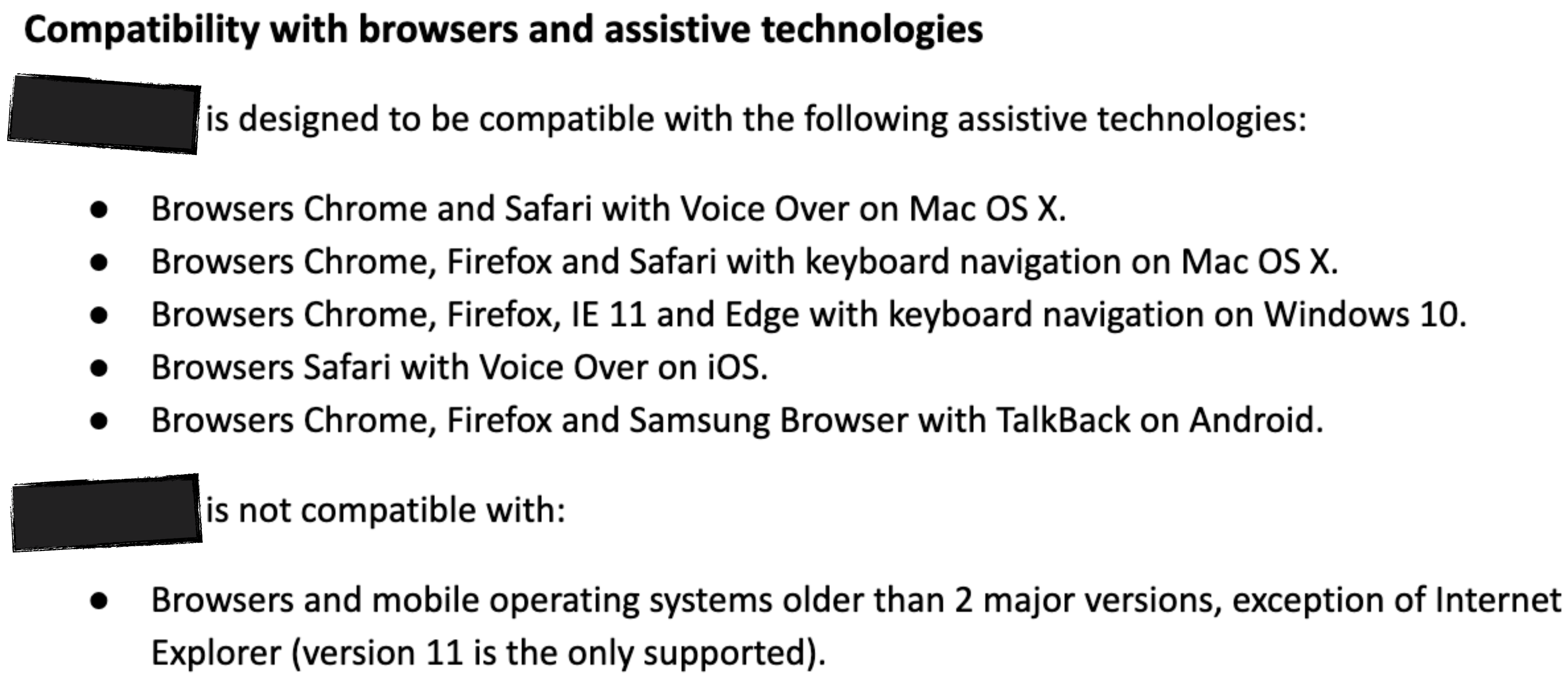
✅ Are all tests successful? Any limitation?

🏃🏾‍♀️ Was the experience seamless each time?

With your team of champions, draft an accessibility policy.

An accessibility policy is an important document about the goals, what's supported and what's not.

You don’t need a complex document.
You don’t even need a legally binding document.
You just need to align, and document your goals.



Compatibility with browsers and assistive technologies

[redacted] is designed to be compatible with the following assistive technologies:

- Browsers Chrome and Safari with Voice Over on Mac OS X.
- Browsers Chrome, Firefox and Safari with keyboard navigation on Mac OS X.
- Browsers Chrome, Firefox, IE 11 and Edge with keyboard navigation on Windows 10.
- Browsers Safari with Voice Over on iOS.
- Browsers Chrome, Firefox and Samsung Browser with TalkBack on Android.

[redacted] is not compatible with:

- Browsers and mobile operating systems older than 2 major versions, exception of Internet Explorer (version 11 is the only supported).

It will make clearer what, when and how to test accessibility.

We all like having a single source of truth.
That’s what this document is for.

And it is a good starting point
for a better design system documentation.

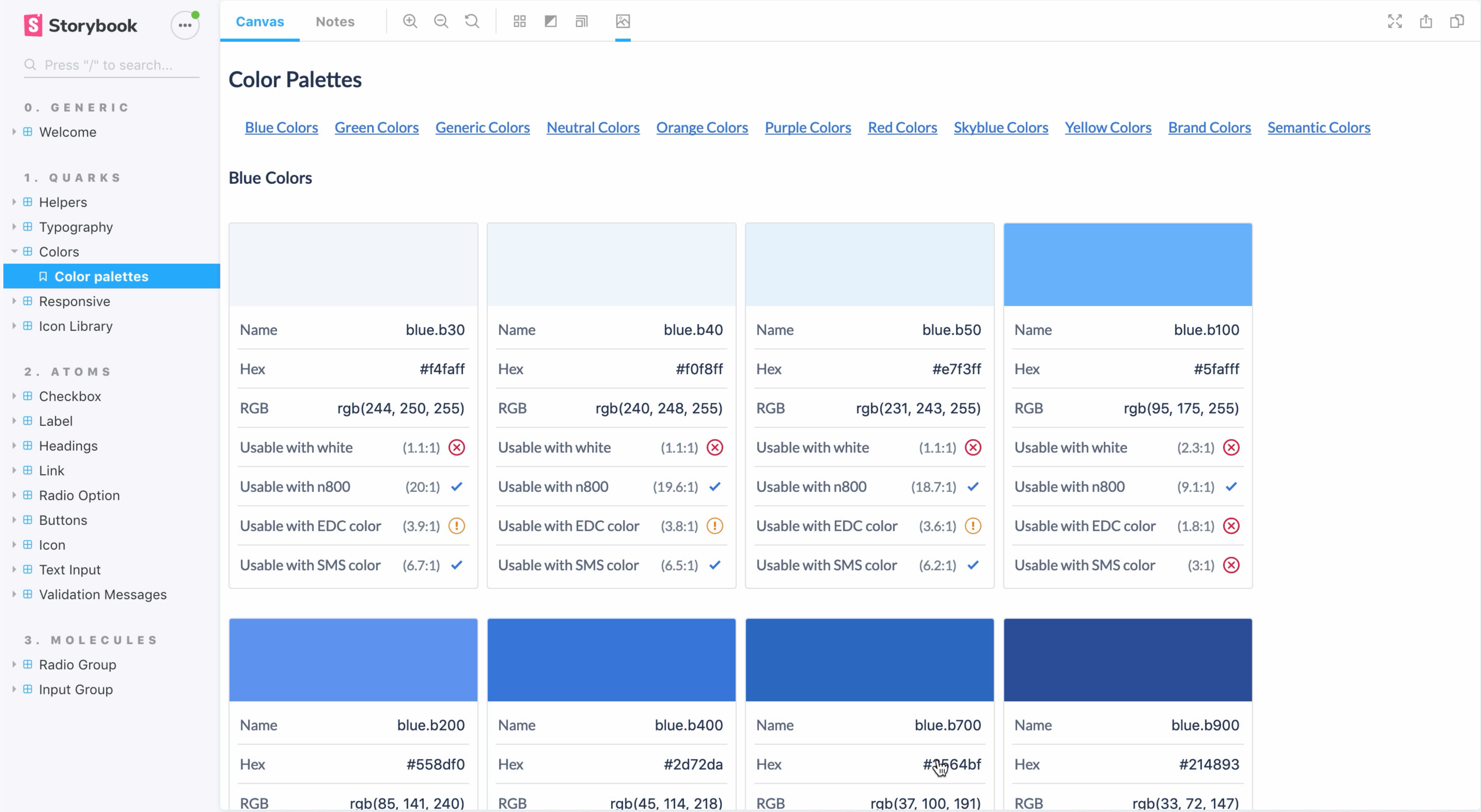
4.

Accessibility & documentation

The Web Content Accessibility Guidelines are **not really the most readable** document ever.

Your documentation should give **context-aware guidance** on how to deliver an accessible product.

Let’s talk about design documentation



0. GENERIC
- Welcome

1. QUARKS
- Helpers
- Typography
- Colors
  - Color palettes
- Responsive
- Icon Library

2. ATOMS
- Checkbox
- Label
- Headings
- Link
- Radio Option
- Buttons
- Icon
- Text Input
- Validation Messages

3. MOLECULES
- Radio Group
- Input Group

Color Palettes

Blue Colors Green Colors Generic Colors Neutral Colors Orange Colors Purple Colors Red Colors Skyblue Colors Yellow Colors Brand Colors Semantic Colors

Blue Colors

Name	blue.b30
Hex	#f4faff
RGB	rgb(244, 250, 255)
Usable with white	(1.1:1) ⊗
Usable with n800	(20:1) ✓
Usable with EDC color	(3.9:1) ⚠
Usable with SMS color	(6.7:1) ✓

Name	blue.b40
Hex	#f0f8ff
RGB	rgb(240, 248, 255)
Usable with white	(1.1:1) ⊗
Usable with n800	(19.6:1) ✓
Usable with EDC color	(3.8:1) ⚠
Usable with SMS color	(6.5:1) ✓

Name	blue.b50
Hex	#e7f3ff
RGB	rgb(231, 243, 255)
Usable with white	(1.1:1) ⊗
Usable with n800	(18.7:1) ✓
Usable with EDC color	(3.6:1) ⚠
Usable with SMS color	(6.2:1) ✓

Name	blue.b100
Hex	#5fafff
RGB	rgb(95, 175, 255)
Usable with white	(2.3:1) ⊗
Usable with n800	(9.1:1) ✓
Usable with EDC color	(1.8:1) ⊗
Usable with SMS color	(3:1) ⊗

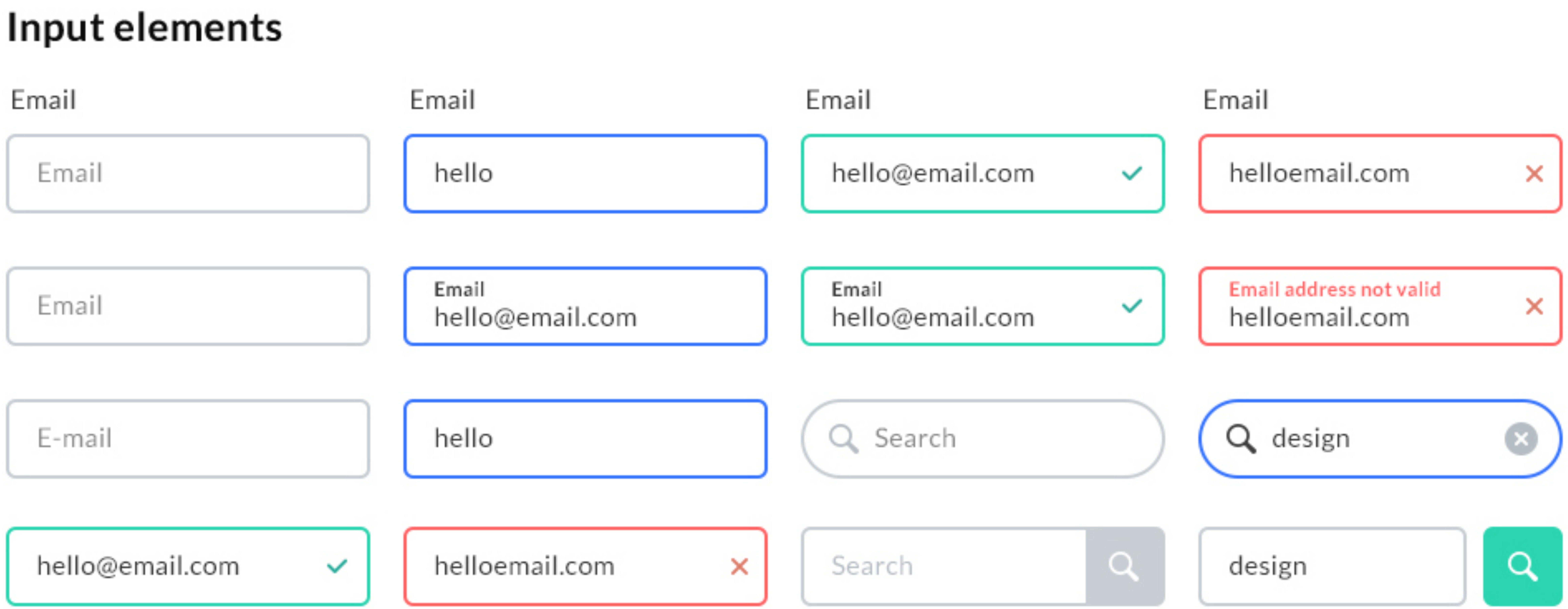
Name	blue.b200
Hex	#558df0
RGB	rgb(85, 141, 240)

Name	blue.b400
Hex	#2d72da
RGB	rgb(45, 114, 218)

Name	blue.b700
Hex	#2564bf
RGB	rgb(37, 100, 191)

Name	blue.b900
Hex	#214893
RGB	rgb(33, 72, 147)

Each component should **support** and **showcase** all possible **state**.



Input elements

Email

Email

Email

Email

Email

hello

hello@email.com ✓

helloemail.com ✕

Email

Email
hello@email.com

Email
hello@email.com ✓

Email address not valid
helloemail.com ✕

E-mail

hello

Search

design

hello@email.com ✓

helloemail.com ✕

Search

design

A library of form input elements within the design system.

butterfly.com.au

You should provide product-specific guidelines.

P
Product
Overview
FOUNDATIONS
Accessibility
Color
Grid
Iconography
Typography
Writing style
COMPONENTS
Avatars
Badges
Banners
Breadcrumbs
Buttons
Checkboxes
Code
Date/time picker
Dropdown menu
Flag messages
Inline dialog
1

Accessibility

Products and web properties should be accessible to everyone, including those with vision, hearing, cognitive, or motor impairments. Accessible design lets people of all abilities interact with, understand, and navigate our products.

Keep in mind that:

- Accessibility is about having a good understanding of our user's journeys and proactively anticipating their needs.
- Accessible design is everyone's responsibility, from information and user experience design, through development, and on into help and support.

Structure & hierarchy

Consistent, clear hierarchy helps users who navigate the page using links or headers. Use headings and titles to outline the page so that users can see the structure and how the sections relate. Give users feedback so they know where they are in the application.

List by level of importance

Place items on the page in order of their level of importance so that users don't have to search for them.

P Product

Overview

FOUNDATIONS

Accessibility

Color

Grid

Iconography

Typography

Writing style

COMPONENTS

Avatars

Badges

Banners

Breadcrumbs

Buttons

Checkboxes

Code

Date/time picker

Dropdown menu

Flag messages

Inline dialog

Structure & hierarchy

Consistent, clear hierarchy helps users who navigate the page using links or headers. Use headings and titles to outline the page so that users can see the structure and how the sections relate. Give users feedback so they know where they are in the application.

List by level of importance

Place items on the page in order of their level of importance so that users don't have to search for them.

Have a consistent hierarchy

Create and maintain a consistent hierarchy so that users can use alternative input methods to move through the page. Headings are in order without skipping levels.

Product
Overview
FOUNDATIONS
Accessibility
Color
Grid
Iconography
Typography
Writing style
COMPONENTS
Avatars
Badges
Banners
Breadcrumbs
Buttons
Checkboxes
Code
Date/time picker
Dropdown menu
Flag messages
Inline dialog

Have a consistent hierarchy

Create and maintain a consistent hierarchy so that users can use alternative input methods to move through the page. Headings are in order without skipping levels.

Do: H1, H2, H3, H3, H4

H1
H2
Paragraph
H3
H4
H5
H6
H7

Don't: H2, H1, H3, H3

Text scaling

Test the UI with color correction, magnification, and other accessibility settings to ensure the layout works with assistive settings.

P
Product
Overview
FOUNDATIONS
Accessibility
Color
Grid
Iconography
Typography
Writing style
COMPONENTS
Avatars
Badges
Banners
Breadcrumbs
Buttons
Checkboxes
Code
Date/time picker
Dropdown menu
Flag messages
Inline dialog
1
2
3
4
5
6
7
8
9
H1
H2
Paragraph
H3
H4
H5
H6
H7

- Tell where the keyboard focus is

Manage focus

Keyboard focus follows the page as the eye would scan it. Focus travels top to bottom, left to right, moving from most to least important item. Users can navigate applications using alternative input methods (D-pads, trackballs, keyboards, and navigation gestures), and the focus flows in a logical order.

- Be conscious of the order of elements on the page. Do they make sense from the user's perspective?
- Indicate where focus is.
- Don't use input focus to select, trigger events, or display messages. Users may need to navigate through all controls before making changes.
- Keep in mind where the focus moves when the element in focus disappears. Users should return to wherever they were before they focused on the element.

Use tooltips

P
Product
Overview
FOUNDATIONS
Accessibility
Color
Grid
Iconography
Typography
Writing style
COMPONENTS
Avatars
Badges
Banners
Breadcrumbs
Buttons
Checkboxes
Code
Date/time picker
Dropdown menu
Flag messages
Inline dialog

Validate forms in-line

Validate forms in-line so keyboard users don't have to navigate far to get feedback.

Form

Text field

@#$%^&

You must enter a valid text

Selects

You must select something

Text field

@#$%^&

This will not be shared with others

Sometimes these help texts can be super long and verbose. The general rule of thumb is to keep these as short as possible while also being explicit. Otherwise people will end up with errors that jump around and are impossible to rea...

Submit

Meaningful text

Consistent and helpful text makes the user interface accessible to users who use a screen reader. Screen readers help users with visual impairments by reading both visible and non-visible alternative text aloud.

All text should support accessibility, whether it's visible (UI labels, headings, buttons, forms

P
Product
Overview
FOUNDATIONS
Accessibility
Color
Grid
Iconography
Typography
Writing style
COMPONENTS
Avatars
Badges
Banners
Breadcrumbs
Buttons
Checkboxes
Code
Date/time picker
Dropdown menu
Flag messages
Inline dialog

Include visual cues

Don't convey information using color alone. Use multiple visual cues, such as stroke weight, patterns, shape, text, or illustrations to ensure that all users receive the same information.

This helps users who are unable to, or have difficulty with, distinguishing one color from another. This includes people who are color blind, have low vision, or are blind.

For example, these inline validation messages use both color and icons to distinguish severity:

Password

••••••••••••••••

✓ Very strong

Email*

jrota

! You need to provide a valid email address

Use high contrast

High color contrast helps users who are partially or completely color-blind see differences between certain colors. It creates a strong visual hierarchy and improves usability for everyone. Make sure that the combination of text and background color do not fall below the WCAG recommended threshold ratio of 4.5:1 for standard text and 3:1 for larger text.

Decorative images and disabled states don't have contrast requirements.

Do	Don't
CRITICAL LOZENGE	CRITICAL LOZENGE

Having a page with all information can quickly be over-whelming and difficult to maintain.

Prefer accessibility requirements per components: be context-aware and actionable.

salesforce
Lightning Design System
Search
What's New
Getting Started
Platforms
Design Guidelines
Accessibility
Component Blueprints
Overview
Accordion
Activity Timeline
Alert
App Launcher
Avatar
Avatar Group
Badges
Brand Band
Breadcrumbs
Sections
About Form Elements
Structure
Base
Form Label
Form Control
States
View Mode/Static
Inline Edit
Help Text Icon
Showing tooltip
Feedback
Required
Error
Layout
Stacked
Horizontal
Single Column Support
Standalone
With 50/50 split
Compound
Rows
Fields
Address
Usage Examples
Record Form
View Mode
Stacked Alignment
Horizontal Alignment
Edit Mode
Stacked Alignment
Show Code

```
<div class="slds-form-element">
  <label class="slds-form-element__label" for="form-element-03">
    <abbr class="slds-required" title="required">* </abbr>Form Label</label>
```

Error

If an error has occurred while submitting a form, the form element with an error should provide feedback. The `slds-has-error` class is placed on the `<div class="slds-form-element">` element. Then, the error message for the user is placed in a `<div>` with the `slds-form-element__help` class.

Accessibility requirement

When a form element displays feedback notifying the user of an error, the error string should be linked to the element by adding the **`aria-describedby`** attribute to the **`<input>`**. The **`aria-describedby`** attribute must reference the id of the error message. This configuration allows screen readers to read the associated error message when the invalid field is focused.

* Form Label

Placeholder Text

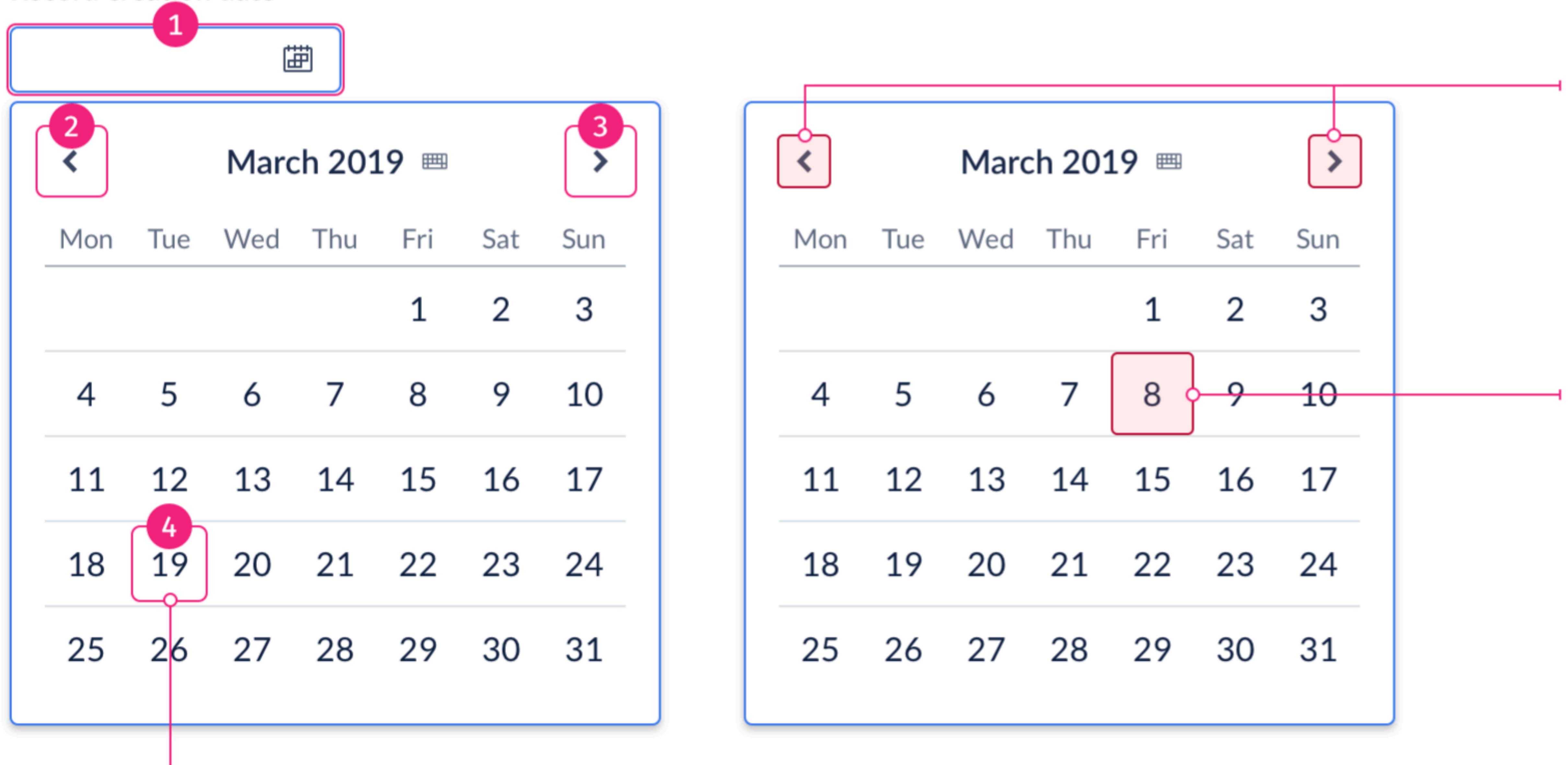
This field is required

```
<div class="slds-form-element slds-has-error">
  <label class="slds-form-element__label" for="form-element-05">
    <abbr class="slds-required" title="required">* </abbr>Form Label</label>
```

It can be a set of technical requirements

Default date picker — Single day selection

Record creation date

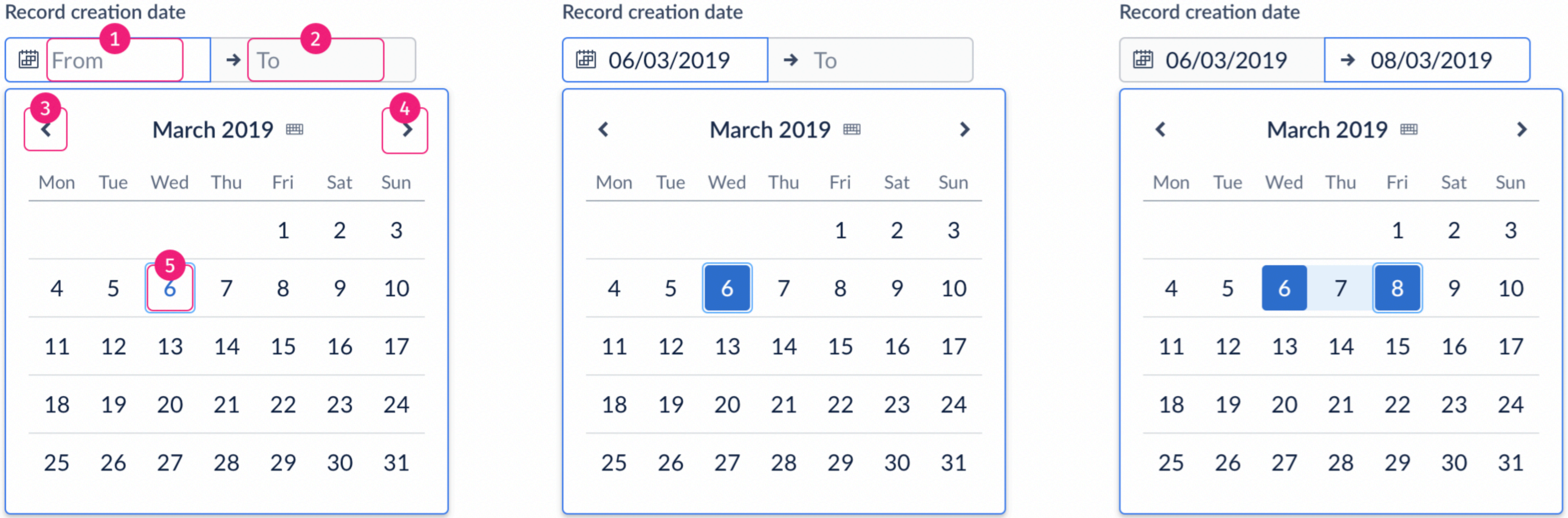


Interactive area:
Loads the previous/next month without transition.

Interactive area:
Fills the input with the selected day and close the date picker.

Focuses on current day if there is no value.
Focuses on entered day if there is value.
Navigation within the numbers by using the arrow keys.
Loads prev/next month when going out of current month.
Page up / Page down to load the next / previous month.
Home / End to move quickly to the first / last day of the month.

But also some requirements regarding the user experience



Record creation date

From → To

March 2019

Mon	Tue	Wed	Thu	Fri	Sat	Sun
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Record creation date

06/03/2019 → To

March 2019

Mon	Tue	Wed	Thu	Fri	Sat	Sun
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

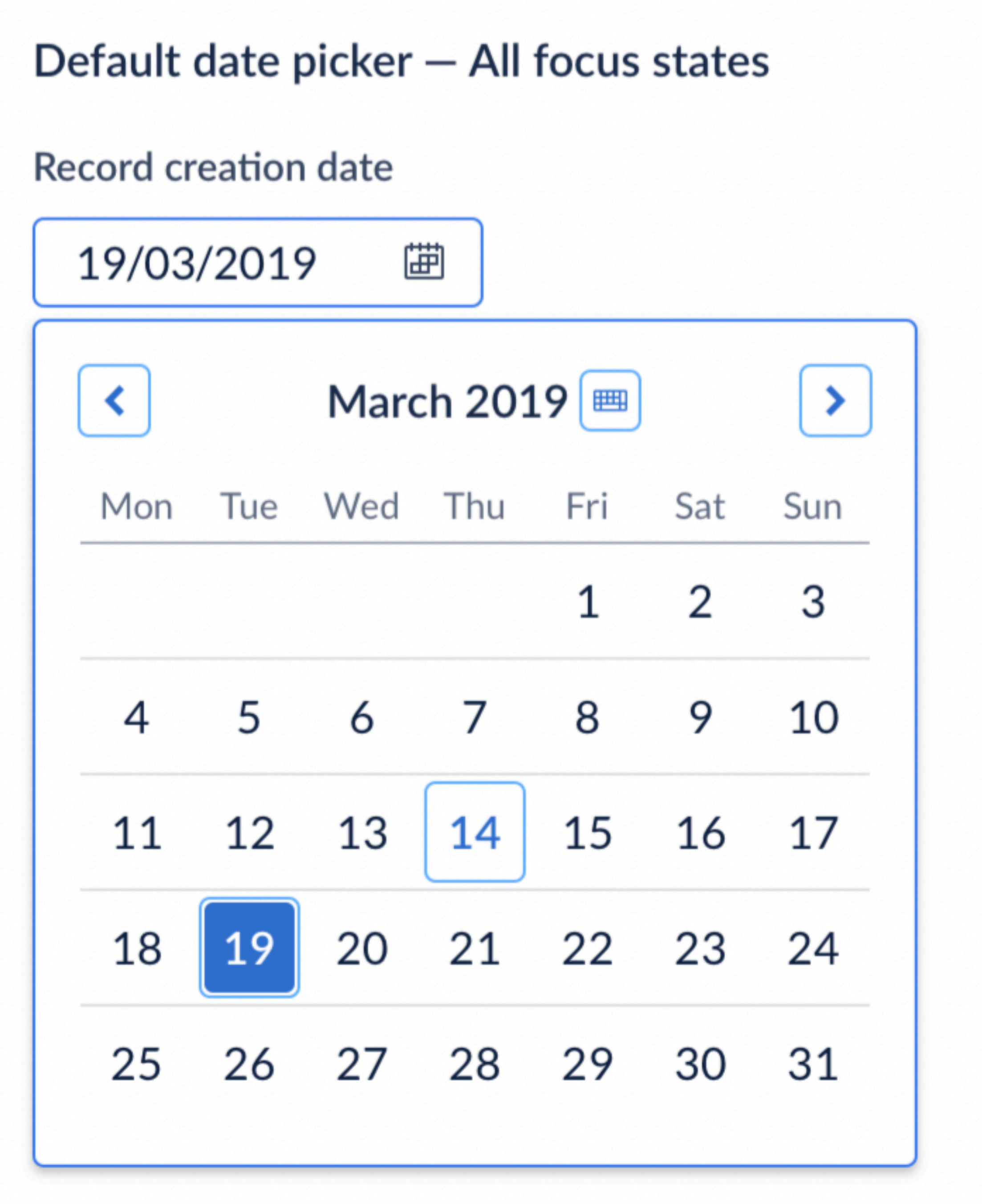
Record creation date

06/03/2019 → 08/03/2019

March 2019

Mon	Tue	Wed	Thu	Fri	Sat	Sun
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

We always think the experience for pointers, but what about keyboards?



Default date picker — All focus states

Record creation date

19/03/2019

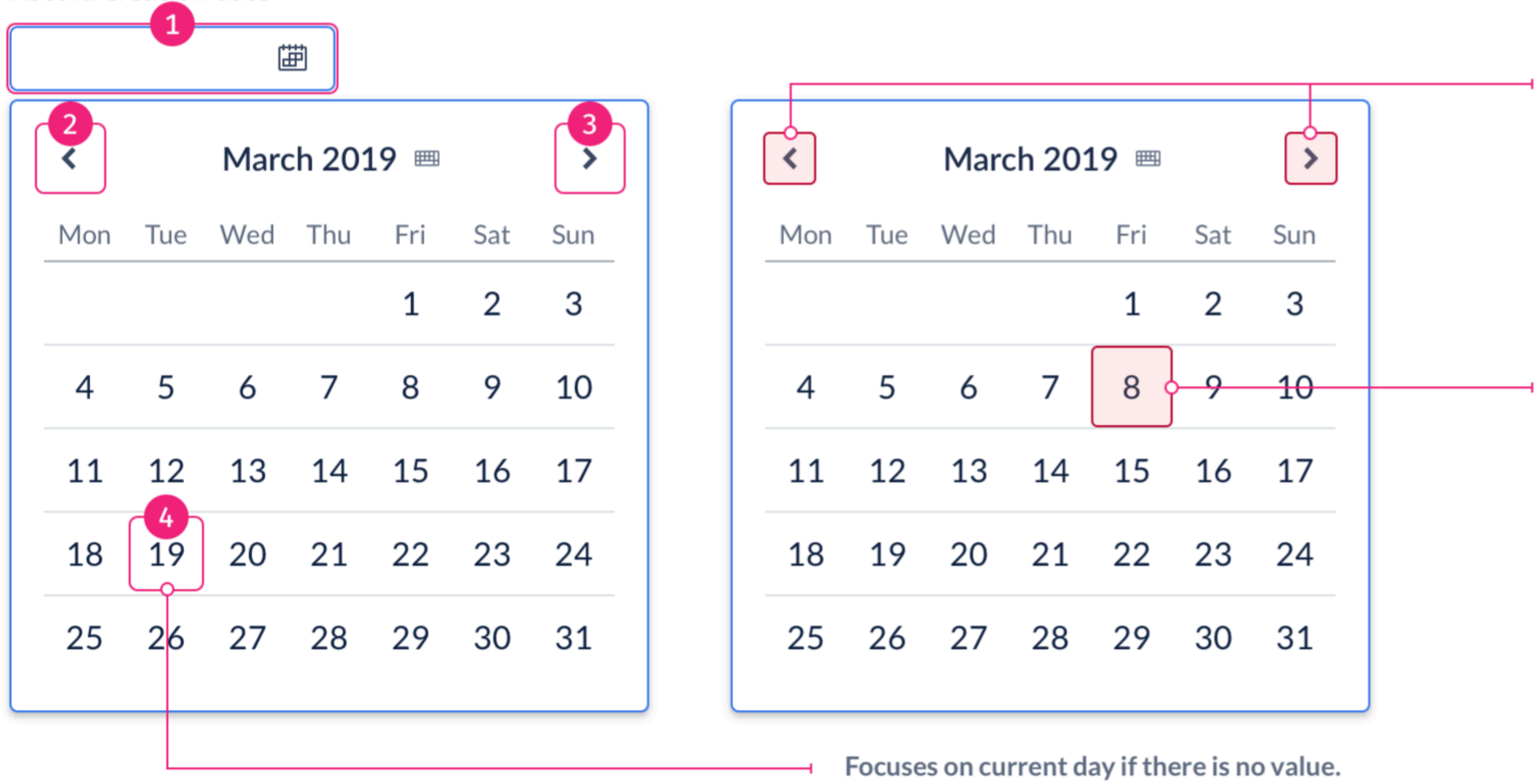
March 2019

Mon	Tue	Wed	Thu	Fri	Sat	Sun
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

If you design hover states, you definitely have time for focus states!

Default date picker — Single day selection

Record creation date



March 2019

Mon	Tue	Wed	Thu	Fri	Sat	Sun
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

March 2019

Mon	Tue	Wed	Thu	Fri	Sat	Sun
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

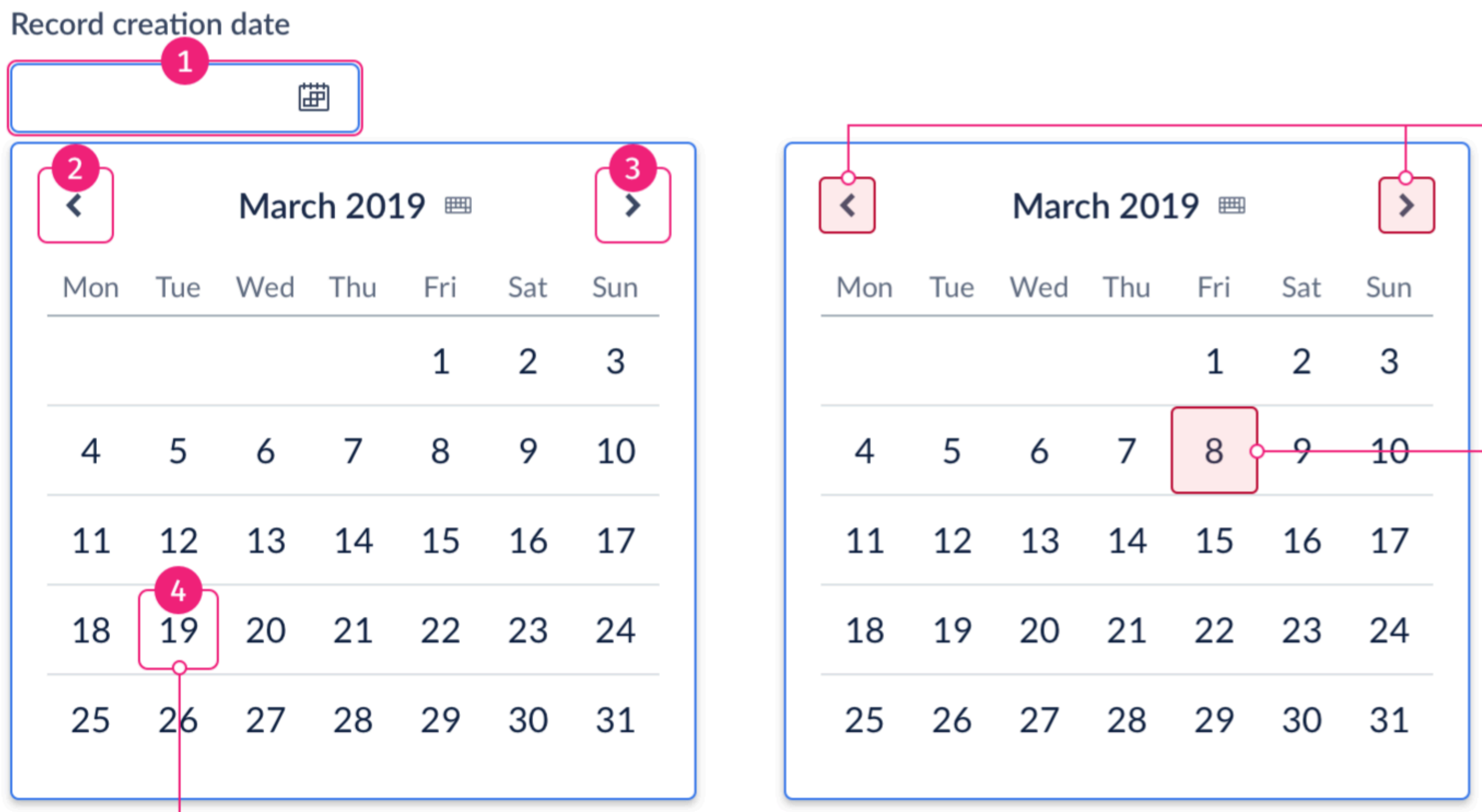
Interactive area:
Loads the previous/next month without transition.

Interactive area:
Fills the input with the selected day and close the date picker.

Focuses on current day if there is no value.
Focuses on entered day if there is value.
Navigation within the numbers by using the arrow keys.
Loads prev/next month when going out of current month.
Page up / Page down to load the next / previous month.
Home / End to move quickly to the first / last day of the month.

Also, remember that keyboard navigation is accessible, but also good for power users looking for more productivity

Default date picker — Single day selection



Record creation date

March 2019

Mon	Tue	Wed	Thu	Fri	Sat	Sun
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

March 2019

Mon	Tue	Wed	Thu	Fri	Sat	Sun
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Interactive area:
Loads the previous/next month without transition.

Interactive area:
Fills the input with the selected day and close the date picker.

Focuses on current day if there is no value.
Focuses on entered day if there is value.
Navigation within the numbers by using the arrow keys.
Loads prev/next month when going out of current month.
Page up / Page down to load the next / previous month.
Home / End to move quickly to the first / last day of the month.

This kind of work is also a good opportunity to collaborate between designers and engineers to define everything invisible.

5.

Any good examples?

OUR PHILOSOPHY
ABOUT THIS PROJECT

Accessibility Guidelines

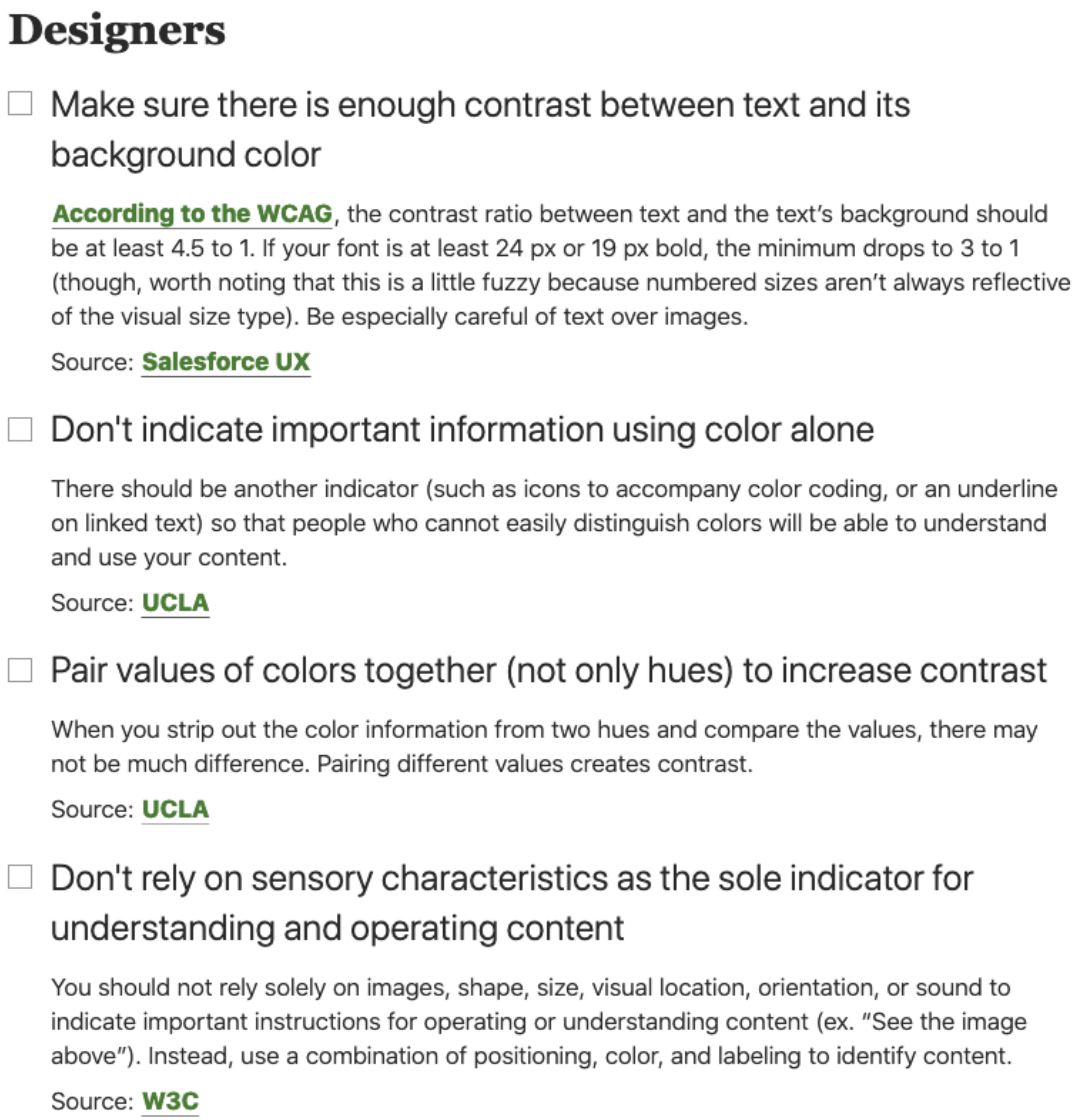
Making work accessible creates a better experience across the board. Use this checklist to help build accessibility into your process no matter your role or stage in a project.

The Checklist

Check the boxes of the guidelines that apply to your project
Preview and copy the checklist to your clipboard
Paste the checklist into a document, Slack, Trello—or however else your team organizes projects

Designers

□ Make sure there is enough contrast between text and its



Designers

□ Make sure there is enough contrast between text and its background color

According to the WCAG, the contrast ratio between text and the text's background should be at least 4.5 to 1. If your font is at least 24 px or 19 px bold, the minimum drops to 3 to 1 (though, worth noting that this is a little fuzzy because numbered sizes aren't always reflective of the visual size type). Be especially careful of text over images.

Source: **Salesforce UX**

□ Don't indicate important information using color alone

There should be another indicator (such as icons to accompany color coding, or an underline on linked text) so that people who cannot easily distinguish colors will be able to understand and use your content.

Source: **UCLA**

□ Pair values of colors together (not only hues) to increase contrast

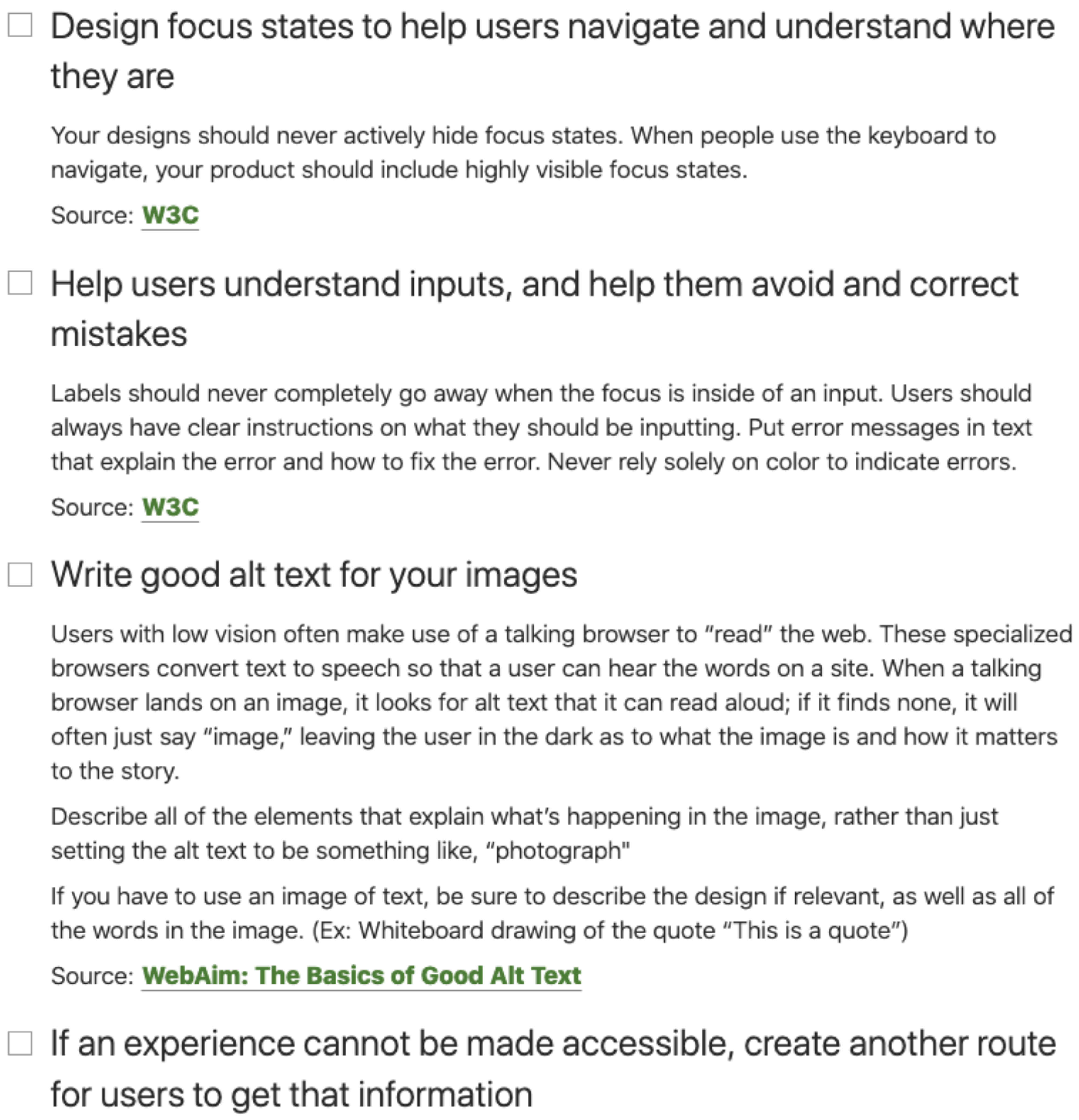
When you strip out the color information from two hues and compare the values, there may not be much difference. Pairing different values creates contrast.

Source: **UCLA**

□ Don't rely on sensory characteristics as the sole indicator for understanding and operating content

You should not rely solely on images, shape, size, visual location, orientation, or sound to indicate important instructions for operating or understanding content (ex. "See the image above"). Instead, use a combination of positioning, color, and labeling to identify content.

Source: **W3C**



□ **Design focus states to help users navigate and understand where they are**

Your designs should never actively hide focus states. When people use the keyboard to navigate, your product should include highly visible focus states.

Source: **W3C**

□ **Help users understand inputs, and help them avoid and correct mistakes**

Labels should never completely go away when the focus is inside of an input. Users should always have clear instructions on what they should be inputting. Put error messages in text that explain the error and how to fix the error. Never rely solely on color to indicate errors.

Source: **W3C**

□ **Write good alt text for your images**

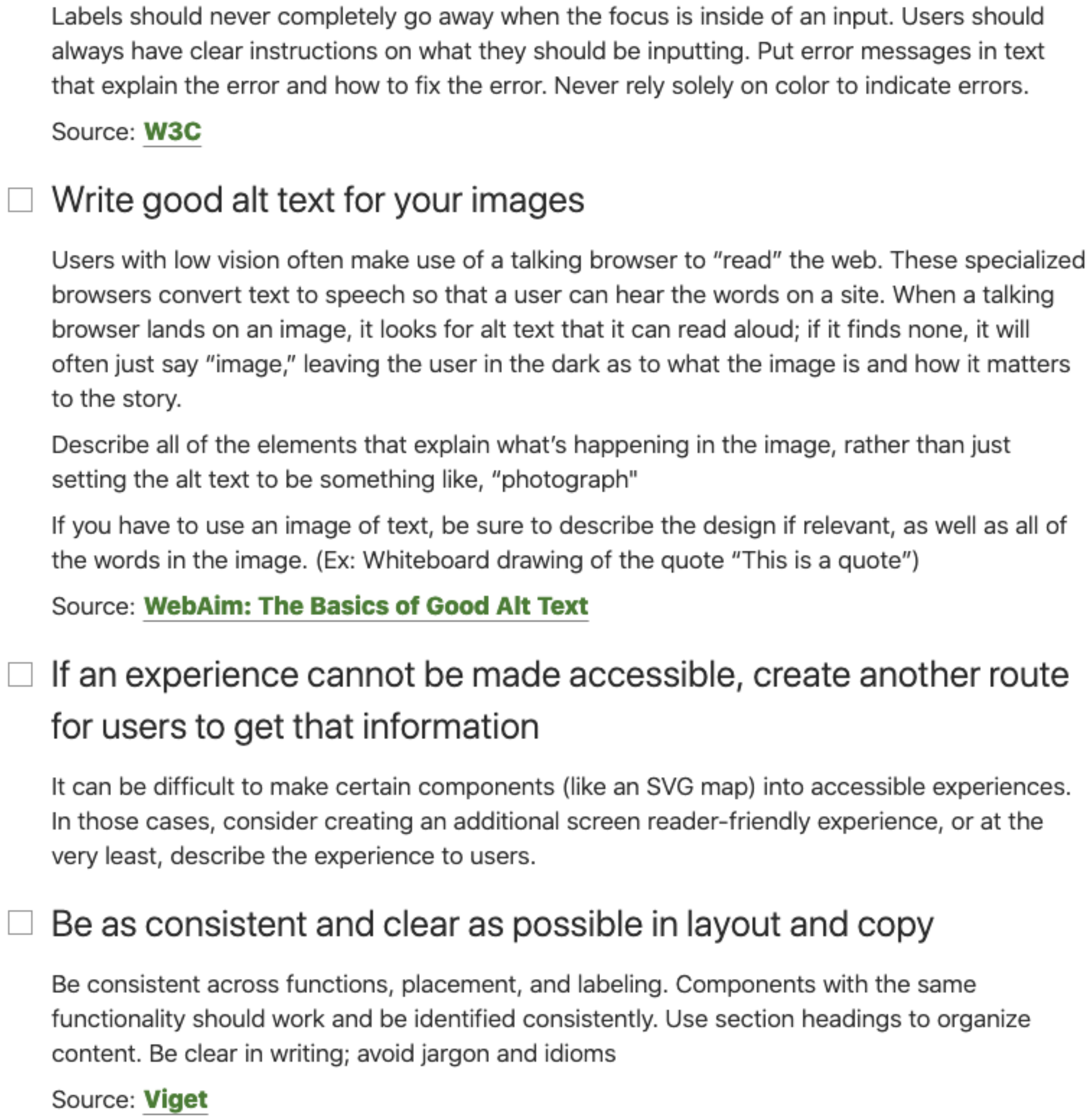
Users with low vision often make use of a talking browser to “read” the web. These specialized browsers convert text to speech so that a user can hear the words on a site. When a talking browser lands on an image, it looks for alt text that it can read aloud; if it finds none, it will often just say “image,” leaving the user in the dark as to what the image is and how it matters to the story.

Describe all of the elements that explain what’s happening in the image, rather than just setting the alt text to be something like, “photograph"

If you have to use an image of text, be sure to describe the design if relevant, as well as all of the words in the image. (Ex: Whiteboard drawing of the quote “This is a quote”)

Source: **WebAim: The Basics of Good Alt Text**

□ **If an experience cannot be made accessible, create another route for users to get that information**



Labels should never completely go away when the focus is inside of an input. Users should always have clear instructions on what they should be inputting. Put error messages in text that explain the error and how to fix the error. Never rely solely on color to indicate errors.

Source: **W3C**

□ Write good alt text for your images

Users with low vision often make use of a talking browser to "read" the web. These specialized browsers convert text to speech so that a user can hear the words on a site. When a talking browser lands on an image, it looks for alt text that it can read aloud; if it finds none, it will often just say "image," leaving the user in the dark as to what the image is and how it matters to the story.

Describe all of the elements that explain what's happening in the image, rather than just setting the alt text to be something like, "photograph"

If you have to use an image of text, be sure to describe the design if relevant, as well as all of the words in the image. (Ex: Whiteboard drawing of the quote "This is a quote")

Source: **WebAim: The Basics of Good Alt Text**

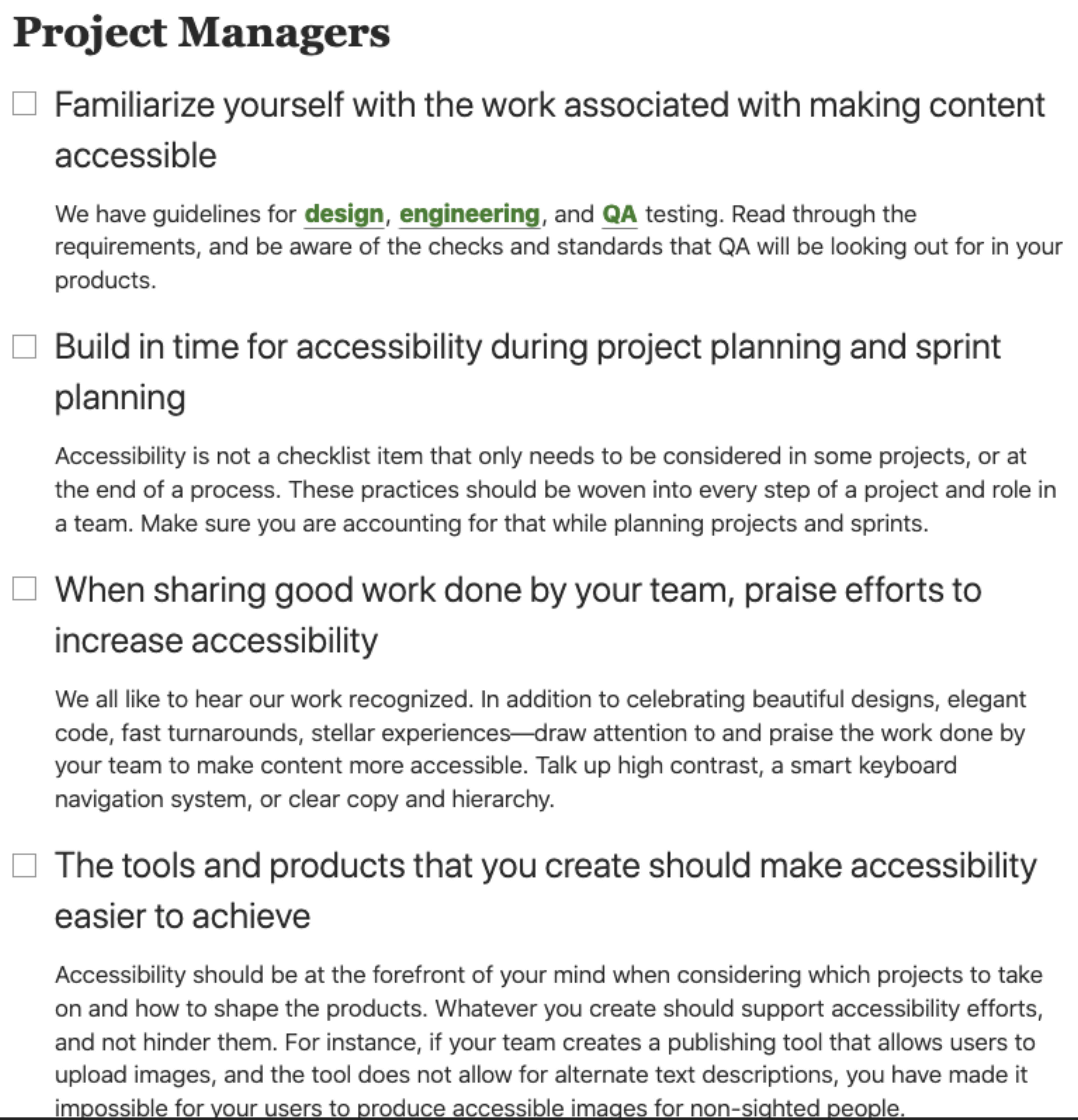
□ If an experience cannot be made accessible, create another route for users to get that information

It can be difficult to make certain components (like an SVG map) into accessible experiences. In those cases, consider creating an additional screen reader-friendly experience, or at the very least, describe the experience to users.

□ Be as consistent and clear as possible in layout and copy

Be consistent across functions, placement, and labeling. Components with the same functionality should work and be identified consistently. Use section headings to organize content. Be clear in writing; avoid jargon and idioms

Source: **Viget**



Project Managers

□ Familiarize yourself with the work associated with making content accessible

We have guidelines for **design**, **engineering**, and **QA** testing. Read through the requirements, and be aware of the checks and standards that QA will be looking out for in your products.

□ Build in time for accessibility during project planning and sprint planning

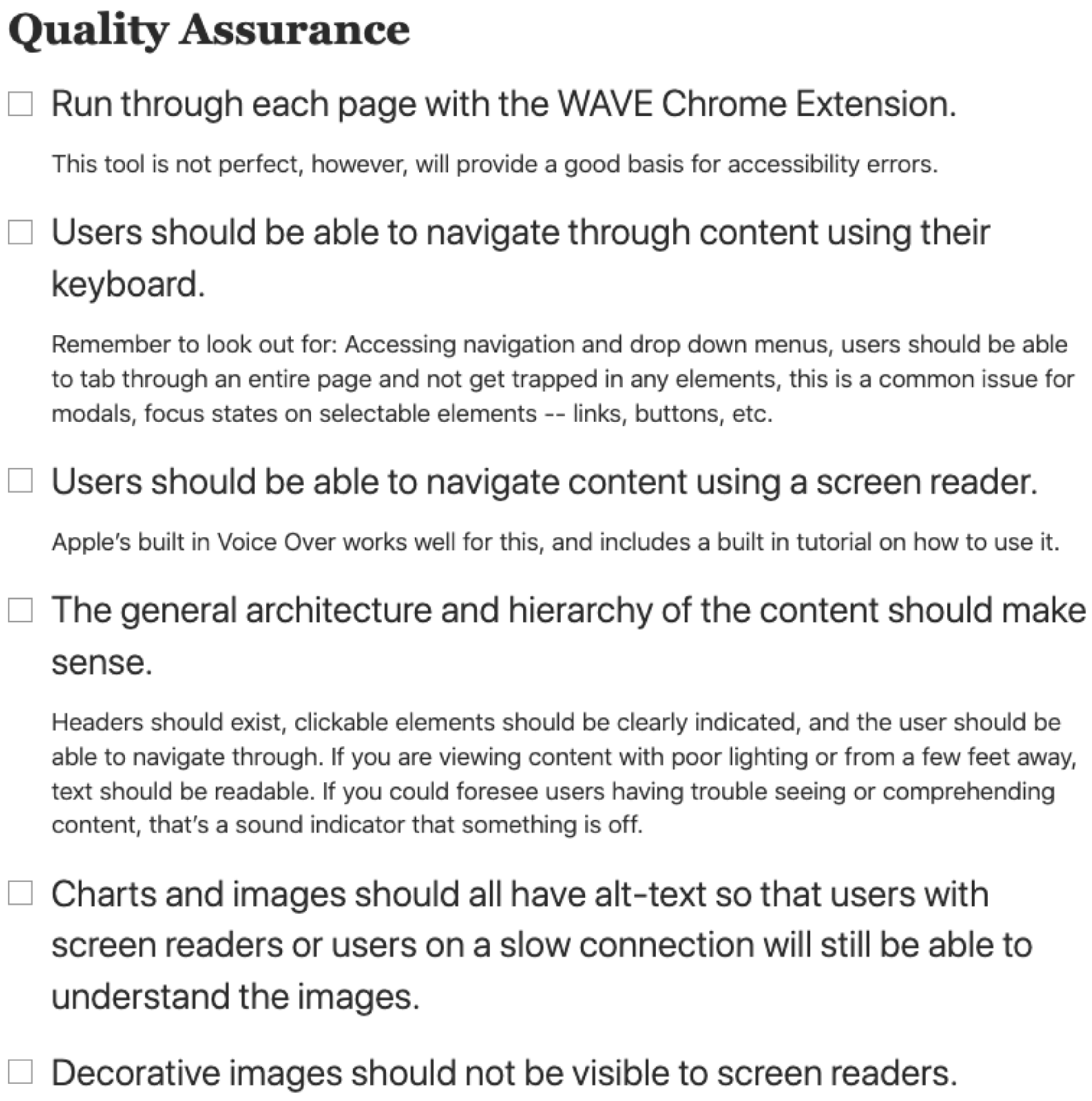
Accessibility is not a checklist item that only needs to be considered in some projects, or at the end of a process. These practices should be woven into every step of a project and role in a team. Make sure you are accounting for that while planning projects and sprints.

□ When sharing good work done by your team, praise efforts to increase accessibility

We all like to hear our work recognized. In addition to celebrating beautiful designs, elegant code, fast turnarounds, stellar experiences—draw attention to and praise the work done by your team to make content more accessible. Talk up high contrast, a smart keyboard navigation system, or clear copy and hierarchy.

□ The tools and products that you create should make accessibility easier to achieve

Accessibility should be at the forefront of your mind when considering which projects to take on and how to shape the products. Whatever you create should support accessibility efforts, and not hinder them. For instance, if your team creates a publishing tool that allows users to upload images, and the tool does not allow for alternate text descriptions, you have made it impossible for your users to produce accessible images for non-sighted people.



Quality Assurance

- □ Run through each page with the WAVE Chrome Extension.

  This tool is not perfect, however, will provide a good basis for accessibility errors.

- □ Users should be able to navigate through content using their keyboard.

  Remember to look out for: Accessing navigation and drop down menus, users should be able to tab through an entire page and not get trapped in any elements, this is a common issue for modals, focus states on selectable elements -- links, buttons, etc.

- □ Users should be able to navigate content using a screen reader.

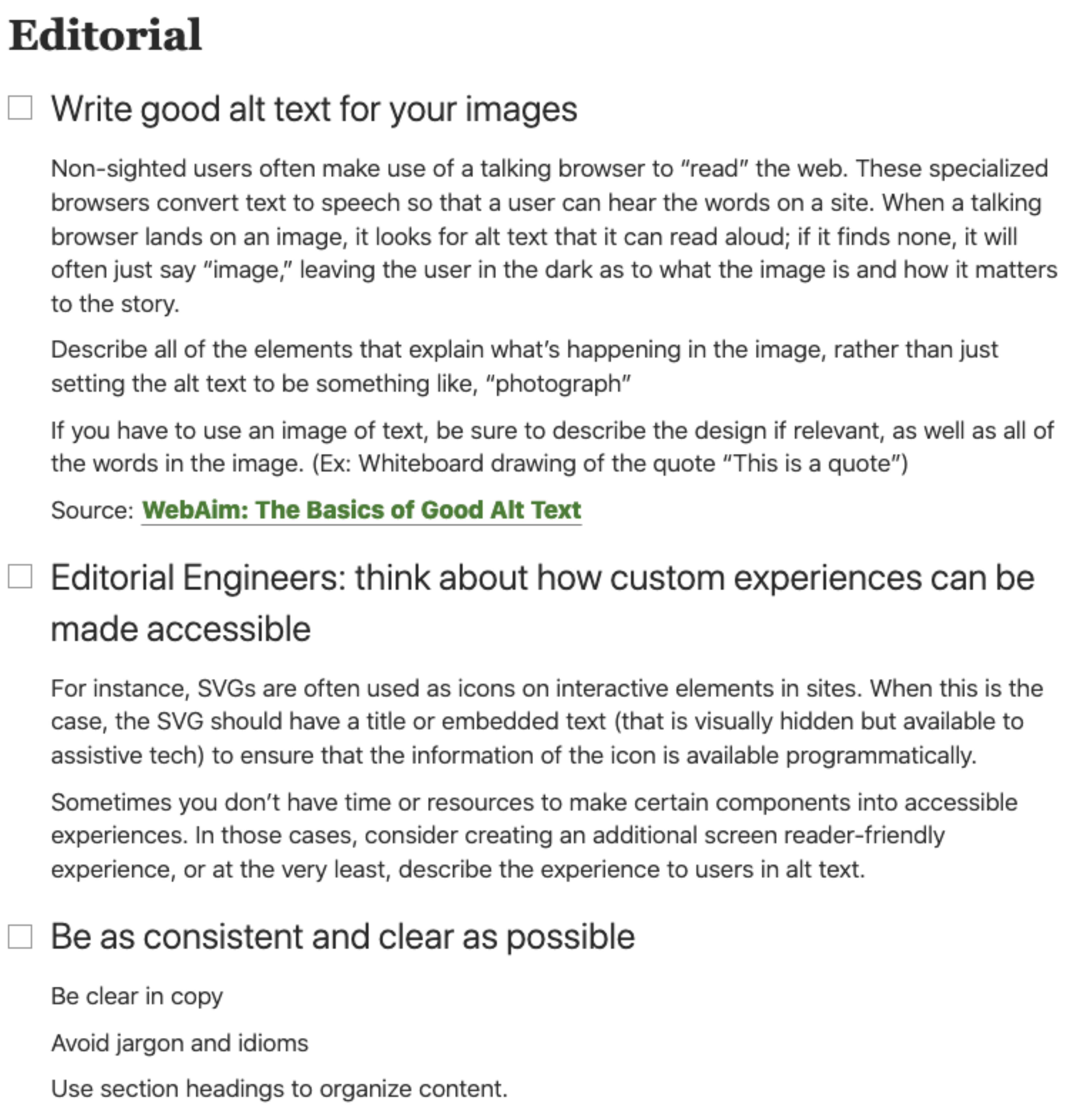
  Apple's built in Voice Over works well for this, and includes a built in tutorial on how to use it.

- □ The general architecture and hierarchy of the content should make sense.

  Headers should exist, clickable elements should be clearly indicated, and the user should be able to navigate through. If you are viewing content with poor lighting or from a few feet away, text should be readable. If you could foresee users having trouble seeing or comprehending content, that's a sound indicator that something is off.

- □ Charts and images should all have alt-text so that users with screen readers or users on a slow connection will still be able to understand the images.

- □ Decorative images should not be visible to screen readers.



Editorial

□ Write good alt text for your images

Non-sighted users often make use of a talking browser to "read" the web. These specialized browsers convert text to speech so that a user can hear the words on a site. When a talking browser lands on an image, it looks for alt text that it can read aloud; if it finds none, it will often just say "image," leaving the user in the dark as to what the image is and how it matters to the story.

Describe all of the elements that explain what's happening in the image, rather than just setting the alt text to be something like, "photograph"

If you have to use an image of text, be sure to describe the design if relevant, as well as all of the words in the image. (Ex: Whiteboard drawing of the quote "This is a quote")

Source: **WebAim: The Basics of Good Alt Text**

□ Editorial Engineers: think about how custom experiences can be made accessible

For instance, SVGs are often used as icons on interactive elements in sites. When this is the case, the SVG should have a title or embedded text (that is visually hidden but available to assistive tech) to ensure that the information of the icon is available programmatically.

Sometimes you don't have time or resources to make certain components into accessible experiences. In those cases, consider creating an additional screen reader-friendly experience, or at the very least, describe the experience to users in alt text.

□ Be as consistent and clear as possible

Be clear in copy

Avoid jargon and idioms

Use section headings to organize content.

Colors
Guidelines
Color palette
Color combinations
Color usage
Accessibility
Resources
Typography
Illustrations
Sounds
Icons
Interaction states
Spacing
Data visualizations

Color combinations

Use the following as a guide for how and when to use colors from our palette together.

Do

Use any light background color with its corresponding text color

Hello

```
background-color: color('orange',
'light');
color: color('orange', 'text');
```

Use any primary background color with white text

Hello

```
background-color: color('blue');
color: color('white');
```

Use white background with any dark text color

Hello

Don’t

Mix any color from the palette with a text color from a different part of the palette.

Hello

```
background-color: color('yellow',
'light');
color: color('blue', 'text');
```

Use low contrast

Hello

```
background-color: color('blue',
'light');
color: color('white');
```

Use black as a background color

Hello

Colors
Guidelines
Color palette
Color combinations
Color usage
Accessibility
Resources
Typography
Illustrations
Sounds
Icons
Interaction states
Spacing
Data visualizations

Accessibility

It's important for Shopify to meet all web accessibility standards. Polaris encourages meeting the minimum contrast ratios specified by WCAG 2.1 Level AA for text, icons, other indicators, and background colors.

✓ Do

Minimal contrast ratio: 4.5

5.25
Grey on Black

4.88
Dark Grey on White

4.87
White on Indigo

6.16
Blue on Blue Light

☒ Don't

Fail

2.94
Dark Grey on Black

2.73
Grey on White

1.70
White on Yellow

1.92
Blue on Blue

Guidelines
2x Grid
Accessibility
Content
Color
Icons
Pictograms
Motion
Spacing
Themes
Typography
Components
Patterns
Experimental
Data visualization
Resources
How to contribute
Updates
Help
Design kit
GitHub

Designing for color blindness

Don’t rely on color alone to convey meaning. This includes conveying information, indicating an action, prompting the user for a response, or distinguishing one visual element from another. When designing with color, it might help to use a color-blind simulator to review visibility of content. If you’re working in Sketch, we recommend the Stark plugin.

Guidelines
2x Grid
Accessibility
Content
Color
Icons
Pictograms
Motion
Spacing
Themes
Typography
Components
Patterns
Experimental
Data visualization
Resources
How to contribute
Updates
Help
Design kit
GitHub

Accessibility

Using various forms of contrast is the most important consideration when making user-friendly color and interface choices. Awareness of standards and best practices is the key to accessible color selection.

The color families in the IBM palette contain ten values from 10 to 100. White and Black sit outside those values. Black text is WCAG AA accessible on colors ranging from 10 to 50. White text is accessible on colors from 60 to 100.

Beyond black and white, the IBM color palette provides a range of accessible combinations. Subtracting the foreground value from the background value (or vice versa) helps determine whether that color combination meets the WCAG AA contrast ratio success criteria. If the difference between two values is 50 or greater, the colors are accessible. Anything below a difference of 50 may fail accessibility standards.

Blue color family, 60 background

For a complete look at color-related accessibility topics in Carbon, visit the Accessibility section.

Search
What's New
Getting Started
Platforms
Design Guidelines
Accessibility
Overview
Guidelines
Patterns
Overview
Accordion
Listbox
Menu
Toggle Button
Tooltip
Component Blueprints
Utilities
Design Tokens
Icons
Downloads

A listbox provides a way to select or reorder options in a predefined list and a selected option can then be made to perform a single action.

Behavior

- Listboxes have options that can be selected, reordered and made to perform an action
- Options themselves **cannot contain** any interactive elements. For example, there cannot be a button, link or input within an option
- Option names are flattened strings and **should not** contain any formatting, semantics or hierarchy that is a requirement for understanding the options meaning
- Options **cannot** be required
- Listboxes can be disabled, where each option is not selectable
- Individual options can also be disabled
- Listbox options can be grouped, with a label for each group

Keyboard Interaction

Single Select Listbox

Option 1
Option 2
Option 3
Option 4

- A listbox should act as a single tab stop, in that a user should be able to press the tab key once to leave the control
- Only one option should be focusable in the list. That option should be the most recently selected option. By default that is the first option in the list
- Up and Down arrows: move **focus and selection** to the previous/next option
- Arrow navigation should wrap within the list when at the first or last option, unless you are using lazy loading or infinite scrolling techniques
- Recommended if more than 5 options:

🎉 In conclusion…

1.

Accessibility
is as fun as frustrating.

2.

Setup an accessibility policy.

3.

Offer ways to learn more about a11y.

4.

Build a team of evangelists.

5.

Propose a documentation adapted to the product

6.

Develop a culture of **test** regardless the job title.

Merci beaucoup ! 🇫🇷

Thank you! 🇬🇧

Vielen Dank! 🇩🇪

Bedankt! 🇳🇱

Tack så mycket! 🇸🇪

Tusen takk! 🇳🇴

@iamhiwelo

Damien Senger

Queer Web worker,
designing design systems and trying to make accessibility more accessible.

raccoon.studio · noti.st/hiwelo

@iamhiwelo